実設計に即した車載LAN

FlexRay 仕様解説

第2回 フレーム・フォーマット

システムの記事

佐藤道夫

前回（本誌2005年4月号，pp.120-127）は，FlexRayを用いたシステム設計において重要となるトポロジとメディア・アクセス方式について説明した．今回は，ネットワーク上でやり取りされる信号（フレーム・フォーマット）について解説する．また，フレーム送信のためにどの程度のスロット長が必要になるかを例を挙げて算出してみる．この計算は，実際の開発ではシステム設計者が行うことになる．（編集部）

前回（本誌2005年4月号，pp.120-127）は，FlexRayを採用したシステムを設計するにあたって，どのようなネットワーク形態を採るのか（トポロジ），どのようなルールにのっとってデータの送信が行われるのか（メディア・アクセス方式）を解説しました．では，実際にネットワーク上ではどのようなメッセージのやり取りが行われるのでしょうか．今回は，このあたりに焦点を当てて，説明していきます．

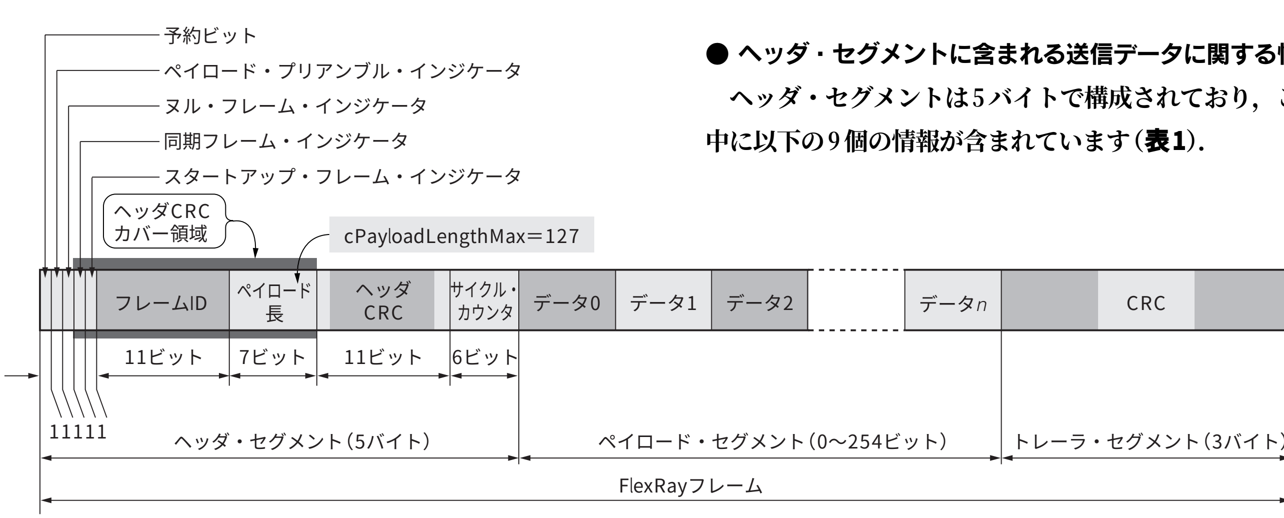


図1　フレーム・フォーマット
FlexRayのフレーム・フォーマットは，ヘッダ・セグメント，ペイロード・セグメント，トレーラ・セグメントの三つのセグメントで構成される．

1 一つのフレームは三つの領域で構成される

FlexRayに対応したネットワーク上では，**図1**に示すようなメッセージ（フレーム）が送信されます．FlexRayのフレームは，次の三つのセグメントで構成されます．

- **ヘッダ・セグメント**
- **ペイロード・セグメント**
- **トレーラ・セグメント**

ヘッダ・セグメントには送信するデータに関する情報が，またペイロード・セグメントには送信すべきデータそのものが格納されています．トレーラ・セグメントはフレーム全体でエラーがあるかどうかをチェックするための機能を備えています．具体的には，ヘッダ・セグメントとペイロード・セグメントをCRC（cyclic redundancy check）にかけた結果が入っています．

● ヘッダ・セグメントに含まれる送信データに関する情報

ヘッダ・セグメントは5バイトで構成されており，この中に以下の9個の情報が含まれています（**表1**）．

表1 ヘッダ・セグメントの内容

名称	値	状態
予約ビット	0	つねに‘0’
ペイロード・プリアンブル・インジケータ	0	ペイロード・セグメントにオプション・ベクタは存在しない
	1	ペイロード・セグメントにネットワーク管理ベクタを含む(静的セグメント)
		垂直方向を中央にペイロード・セグメントにメッセージIDを含む(動的セグメント)
ヌル・フレーム・インジケータ	0	ペイロード・セグメントには無効な(null)データが含まれている
	1	ペイロード・セグメントには有効なデータのみ存在する
同期フレーム・インジケータ	0	そのフレームは同期フレームではない
	1	そのフレームが同期フレームであることを示す
スタートアップ・フレーム・インジケータ	0	そのフレームはスタートアップ・フレームではない
	1	そのフレームがスタートアップ・フレームであることを示す
フレームID	0	無効
	1～2,047	送信するフレームのID番号を示す
ペイロード長	0～127	ここで示された値を2倍すると，ペイロード・サイズになる
ヘッダCRC	0～2,047	同期フレーム・インジケータ，スタートアップ・フレーム・インジケータ，フレームID，ペイロード長で計算されたCRCコードを示す
サイクル・カウンタ	0～63	コミュニケーション・サイクルのカウント値を示す

1）予約ビット(1ビット)

予約ビットは，将来起こりうるプロトコルの変更のために予約されているものです．そのため，実際のFlexRayのアプリケーションには使用できません．送信ノードは予約ビットを論理的に‘0’にセットし，受信ノードはこのビットを無視します．

2）ペイロード・プリアンブル・インジケータ(1ビット)

ペイロード・プリアンブル・インジケータは，送信されたペイロードに特別な機能(オプション・ベクタ)が含まれているかどうかを示します．オプション・ベクタの有無や内容は，ペイロード・プリアンブル・インジケータの値(‘0’か‘1’か)によって決まります．

例えば，ペイロード・プリアンブル・インジケータ=‘0’の場合，ペイロード・セグメントにはオプション・ベクタは含まれていません．

一方，ペイロード・プリアンブル・インジケータ=‘1’の場合，そのフレームが送信されるタイミング(静的セグメントか動的セグメントか)によって，ベクタの内容は異なります．フレームが静的セグメントで送信されていれば，ペイロードの始めにネットワーク管理ベクタ(詳細は後述)が存在することを示しています．フレームが動的セグメントで送信されていれば，ペイロードの始めにメッセージID(詳細は後述)が存在することを示しています．

3）ヌル・フレーム・インジケータ(1ビット)

ヌル・フレーム・インジケータは，フレームのペイロード・セグメントに有用なデータが含まれているかどうかを示すものです．このビットが‘0’の場合，ペイロード・セグメントには無効な(null)データが含まれています．一方，このビットが‘1’であれば，ペイロード・セグメントには有効なデータのみが存在します．

コラム>> パラメータの接頭語について

FlexRay仕様では，“cChannelIdleDelimiter”，“cdBSS”，“gPayloadLengthStatic”などさまざまなパラメータが定められています．これらのパラメータには接頭語が付いており，下記のような意味を持っています．

●c：プロトコル定数

cが頭に付くパラメータは，プロトコルの範囲または特性の定義に使用される値です．

●v：ノード変数

vが頭に付くパラメータは，時間や事象に依存して変化する値です．

●g：クラスタ・パラメータ

gが頭に付くものは，クラスタ内のすべてのノードが同じ値を持たなければならないパラメータです．このパラメータは通信コントローラ自身のコンフィグレーション時に初期化されます．パラメータの変更は通信コントローラがコンフィグレーションするときのみ可能となります．

●p：ノード・パラメータ

pが頭に付くものは，クラスタ内のノードにおいて異なる値を持つことがあるパラメータです．これは，通信コントローラ自身のコンフィグレーション時に初期化され，値の変更もコンフィグレーション時にのみ行うことができます．

●d：継続時間

dが頭に付くパラメータは，継続時間を表します．

4）同期フレーム・インジケータ（1ビット）

同期フレーム・インジケータは，通信フレームが同期フレームかどうかを示します．同期フレーム・インジケータが‘1’の場合はそのフレームを同期フレームとし，すべての受信ノードを同期のために利用します．

5）スタートアップ・フレーム・インジケータ（1ビット）

ノードの電源がONしても，すぐには通信を開始できません．電源をONしてから通信可能な状態になるまでのプロセスを「スタートアップ」と言います（**図2**）．

スタートアップ・フレーム・インジケータが‘1’にセットされているとき，そのフレームがスタートアップ・フレームであることを示します．スタートアップ・フレームは，スタートアップ時において特別な役割を担います．例えば，電源ONのタイミングは各ノードで違うので，クロックの位相も異なりますが，こうしたノード間の同期合わせはスタートアップ・フレームを用いて行います．

このスタートアップ・フレームは，コールド・スタート・ノード（スタートアップ時の初期化を行うためのノード）だけが送信することができます．もう少し細かく言うと，コールド・スタート・ノードの同期フレームでのみ，スタートアップ・フレーム・インジケータが‘1’にセットされます（つまり，スターアップ・フレーム・インジケータが‘1’にセットされたフレームは，同期フレーム・インジケータも‘1’にセットされている）．

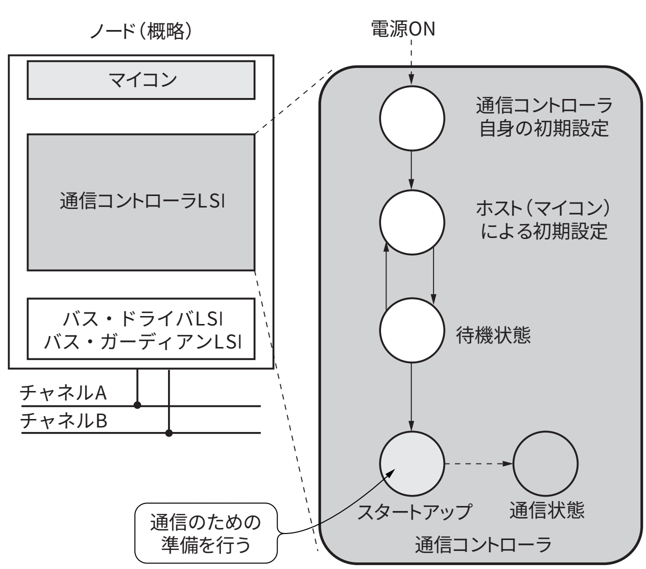


図2　スタートアップ

電源がONになってから通信状態までの状態遷移の概略を示す．通信のための準備を行う状態をスタートアップというが，この状態におけるさまざまな機能はスタートアップ・フレームによって実現される．

6）フレームID（11ビット）

フレームIDは，各スロットで送信すべきフレームを明確にします．フレームIDは，コミュニケーション・サイクル（FlexRayにおける通信の最大単位）で，チャネルごとに1回以上は使用されません．フレームIDのとるべき範囲は1～2,047であり，‘0’の場合は無効となります．フレームIDは，MSB（most significant bit）から送信されます．

7）ペイロード長（7ビット）

ペイロード長には，ペイロード・セグメントのサイズが示されます．具体的には，ペイロード・データのサイズ（バイト数）を2で割った値がここにセットされます．例えば，ペイロード・セグメントが72バイトの場合，このフレームのペイロード長として36がセットされます．

ペイロード長の範囲は，0～127（cPayloadLengthMax）です．このことは，ペイロード・セグメントの最大値が254バイト（＝2×cPayloadLengthMax）であることと一致します．

コミュニケーション・サイクルの静的セグメントで送信されるすべてのフレームのペイロード長は同一です．こうしたフレームのペイロード長にはデータとしてgPayloadLengthStaticがセットされ，送信されます．動的セグメントでは，フレームごとまたはチャネル間でペイロード長が異なる場合があります．動的セグメントにおけるペイロード長の範囲は0～127バイト（pPayloadLengthDynMax＝0-cPayloadLengthMax）です．

フレームIDの場合と同様に，ペイロード長はMSBから送信されます．

8）ヘッダCRC（11ビット）

ヘッダCRCは，同期フレーム・インジケータ，スタートアップ・フレーム・インジケータ，フレームID，ペイロード長で計算されたCRCコードを含んでいます．

なおFlexRayの場合，通信コントローラでは送信フレームのためのヘッダCRCの計算は行われません．送信フレームのヘッダCRCはホスト（マイコン）のアプリケーションによって計算され，スタートアップ時にホストから通信コントローラに提供することが仕様で決められています．

9）サイクル・カウンタ（6ビット）

サイクル・カウンタは，送信ノードが何回データを送信したか，つまりコミュニケーション・サイクルのカウント値を示します．サイクル・カウンタの最大値は63（cCycle

CountMax＝63）であり，これを超えると0に戻って再度カウントアップします．

サイクル・カウントもMSBから送信されます．

● ペイロード・セグメントにはオプション領域がある

ペイロード・セグメントは，0～254バイトのデータを含んでいます．この値はヘッダ・セグメントのペイロード長（cPayloadLengthMax）を2倍にしたものであり，ペイロード・セグメントに格納されるバイト数はつねに偶数となります．

このペイロード・セグメントのバイト・データについては，ヘッダ・セグメントの後の最初のバイトが「データ0」，次が「データ1」というように順番が付けられています．

このペイロード・セグメントの一部をオプションとして利用できます．オプションには，「ネットワーク管理ベクタ」と「メッセージID」が用意されています．

1）ネットワーク管理ベクタ（NMベクタ）

ネットワーク管理ベクタとして，ペイロード・セグメントの最初の0～12バイトを割り当てることが可能です（**図3**）．ネットワーク管理ベクタは，ユーザが定義するネットワーク管理のためのプロトコルを支援するためのもので，例えばクラスタ全体のスタートアップやシャットダウンを決定します．

ネットワーク管理ベクタは，静的セグメントとして送信される場合に割り当てられます．ネットワーク管理ベクタの長さ（何バイト割り当てるか）は，通信コントローラのコンフィグレーション（初期設定）時にホスト（マイコン）によって書き込まれます．ネットワーク管理ベクタの長さを表すパラメータはgNetworkManagementVectorLengthであり，すべての送信ノードにおいてこの値は等しくなくてはなりません．

なお，ネットワーク管理ベクタはバイトのMSBから送信されます．

2）メッセージID

フレームが動的セグメントとして送信される場合のみ，メッセージIDが使用されます．ペイロード・セグメントの最初の2バイトがメッセージIDとして割り当てられています（**図4**）．

このメッセージIDは，送信するデータの内容を明確にするために決められた番号です．この番号は，ホスト上のアプリケーション・ソフトウェアから通信コントローラに書き込まれます．

上述したように，これらのオプションが存在するかどうかは，ヘッダ・セグメントのペイロード・プリアンブル・インジケータによって示されます．

● トレーラ・セグメントではCRCでエラーをチェック

トレーラ・セグメントは，24ビットCRCを含んでいます（**図5**）．CRC領域には，そのフレームのヘッダ・セグメントとペイロード・セグメントを通じて計算されたCRCコードが含まれています．

FlexRayのネットワーク・トポロジがデュアル・チャネル（チャネルA，チャネルB）の場合，ノードはどちらかのチャネルに依存しており，次に示すようにそれぞれ違う初期ベクタを使用してフレームを送信します．

- チャネルAにフレームを送信するノードは，初期ベクタ0xFEDCBAを使用
- チャネルBにフレームを送信するノードは初期ベクタ0xABCDEFを使用

このことから，AチャネルとBチャネルでは初期ベクタの値が反転していることがわかります．これによって，もしAチャネルとBチャネルがまちがって接続された場合，CRCによってエラーをチェックできます（このチェックは

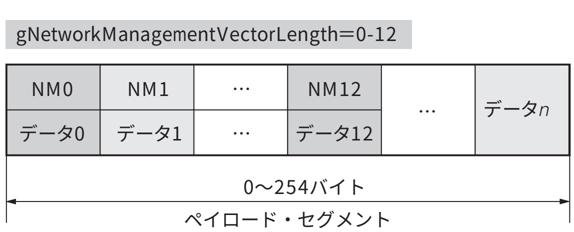


図3　ネットワーク管理ベクタ

ネットワーク管理ベクタとして，ペイロード・セグメントのはじめの0～12バイトまでを割り当てることが可能．ネットワーク管理ベクタは，静的セグメントでのみ使用される．

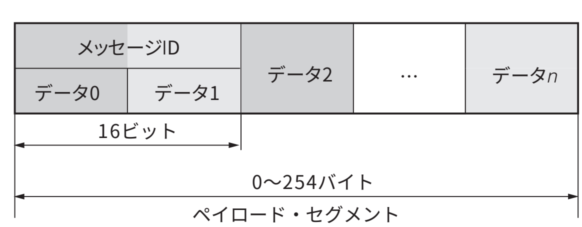


図4　メッセージID

ペイロード・セグメントの最初の2バイトがメッセージIDとして割り当てられる．メッセージIDは，動的セグメントでのみ使用される．

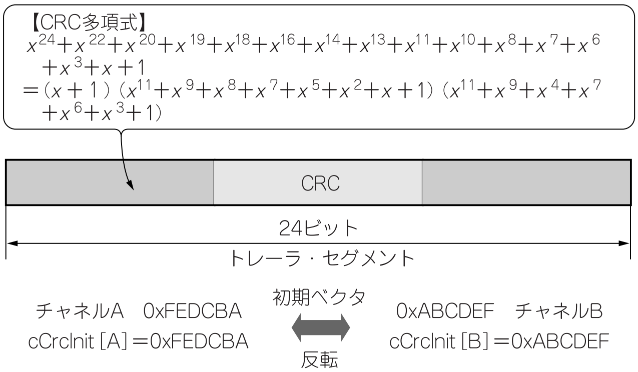


図5　トレーラ・セグメント
トレーラ・セグメントは24ビットCRCを含んでおり，フレーム全体のエラー・チェックの機能を備えている．

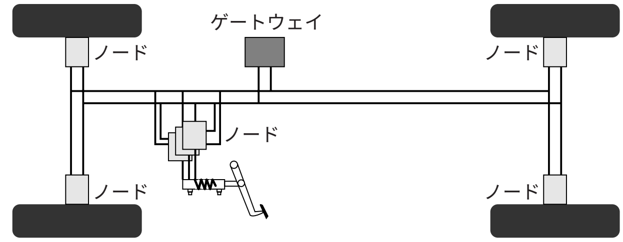


図6　Brake-by-wireのアプリケーション例
本稿では，このブレーキ・システムを例に，メディア・アクセスやフレームについての検討を行う．

アプリケーションに依存する)．

2 メディア・アクセスに対する送信フレームの検討

さて，前回は**図6**に示すBrake-by-wireアプリケーションのアクセス方法を検討しました．そのとき，一つの静的スロットの時間を75MT（マイクロティック）と決めました．今回は，この時間内にどのくらいの長さのペイロードを送信できるかを検討してみましょう．

ところで，筆者はFlexRayのプロトコルを説明する際に，フレーム送信を**図7**の上側のように図示します．静的セグメントでは，スロットの境界とフレームの間に隙間がありますが，これには意味があります．**図7**の下側の図に示すように，フレームの前後にアイドル時間があるからです．

● 送信開始の際のオフセットを設ける

各ノードには必要なスロット数が割り当てられています．また，一つのスロットの送信にかかる時間（スロット時間）もあらかじめ決められています．しかし，その割り当てられた時間のスロットを送信する場合，たとえ準備ができていたとしても通信を開始するまでに時間がかかります．そのため，オフセットが規定されています．FlexRayでは，送信を開始する点を「アクション・ポイント」と言いますが，静的スロットの開始からこのアクション・ポイントまでのオフセットを「アクション・ポイント・オフセット」と呼び，その範囲は1～31MT（gdActionPointOffset＝1-31MT）と規定されています．

アクション・ポイントから送信が開始されるわけですが，実際にはフレームが送信される前に「送信スタート・シーケンス（TSS：Transmission Start Sequence）」に入ります．このシーケンスでは，適正な通信を行う準備のための初期化を行い，これから送信を開始するという合図を送ります．送信スタート・シーケンスは連続した“L”レベル(‘0’)で構成されます．“L”レベルをどの程度の長さにするのかはgdTSSTransmitterというパラメータで割り当てられます．その範囲は1～13ビット（仕様書ではgdBitという単位で表される）です．

この次に，フレーム送信の開始を合図するためのフレーム・スタート・シーケンス（FSS：Frame Start Sequence）に入ります．このシーケンスは，フレームの最初のバイト・スタート・シーケンス（後述）の量子化誤差を補正するためにも使用されます．フレーム・スタート・シーケンスは1gdBitの“H”レベル(‘1’)で構成されます．

フレーム・スタート・シーケンスの次に，バイト・スタート・シーケンス（BSS：Byte Start Sequence）が続きます．このバイト・スタート・シーケンスの立ち下がりエッジがビットの同期に使用されます．

実は，**図8**に示すように，FlexRayのフレーム内のデータは「バイト・スタート・シーケンス＋1バイト」で構成されています（拡張バイト・シーケンスと呼ばれる）．バイト・スタート・シーケンスは，1gdBitの“H”レベルと1gdBitの“L”レベルからなるので，フレームの一つのデータは10ビットで構成されていることになります．

フレームの最後のデータ・バイトの終わりを表す状態をフレーム・エンド・シーケンス（FES：Frame End Sequence）といいます．フレーム・エンド・シーケンスは，1gdBitの“L”レベルと1gdBitの“H”レベルで構成されます．

この後に，11gdBitのチャネル・アイドル境界（アイドル

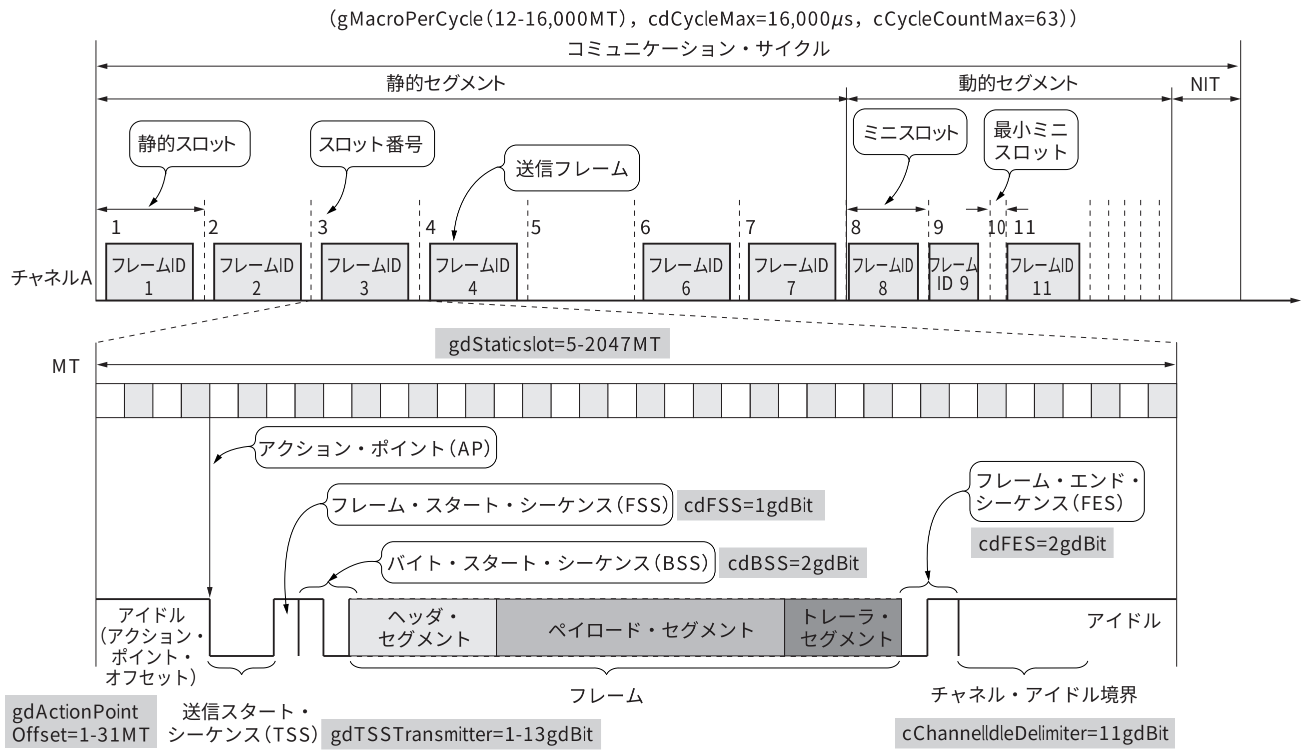


図7　アクション・ポイント・オフセット
FlexRayでは，実際に送信を開始する際の時間のずれを考えて，オフセットを設けている．

を検出するための準備期間）が続き，アイドル状態となり，一つの静的スロットが終了します．

● **静的スロット長にはクロック精度やバス遅延も考慮する**

それでは，静的スロット長を計算してみましょう．

上述のオフセットを含めると，一つの静的スロット長vStaticFrameLength（単位はgdBit）は，**図9**の式(1)で求められます．

ここで，式(1)の中の80という数字は，フレームのヘッダ・セグメントとトレーラ・セグメントにおけるビット(gdBit)数を表しています．ヘッダ・セグメントは5バイト，トレーラ・セグメントは3バイトで固定なので合計8バイトとなります．上述したように，FlexRayのデータ・バイトは実際には10gdBit（8ビットのデータ＋2ビットのバイト・スタート・シーケンス）で構成されるため，8バイト×10gdBit＝80gdBitとなります．

ペイロード・セグメントの場合も同様に，1データ・バイトは10gdBitです．静的セグメントの場合，ペイロード・セグメントにはヘッダ・セグメントのペイロード長フィールドにセットされたパラメータgPayloadLengthStaticの2

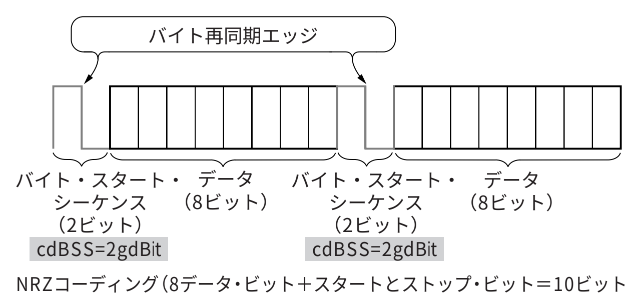


図8　バイト・コーディング
FlexRayのデータは1バイト＋2ビット（バイト・スタート・シーケンス）で構成されている．

倍のデータが入ります．そのため，式(1)ではgPayloadLengthStatic×20(2×10gdBit)となっているわけです．

また，式(1)の二つのパラメータcdFSSとcdFESは，**図7**からわかるように固定値であり，それぞれcdFSS＝1gdBit，cdFES＝2gdBitです．

なお，最大フレーム長はクロックの精度（cClockDeviationMax＝±0.0015）とマクロティックの最小時間によって変化します．

このほかにも，クロック精度の影響や伝播遅延は**図7**に示したようにgdActionPointOffsetとして考慮します．ま

●静的フレーム長

$$\mathrm{vStaticFrameLength} = \mathrm{gdTSSTransmitter} + \mathrm{cdFSS} + 80 + \mathrm{gPayloadLengthStatic} \times 20 + \mathrm{cdFES} \quad \cdots\cdots (1)$$

※単位はすべてgdBit

●サイレント部の長さ

$$\begin{aligned}\mathrm{vSilencePart}[\mathrm{MT}] = \max^{*1}(&\mathrm{gdBGpostEnablePart}[\mathrm{MT}]^{*2} + 1[\mathrm{MT}]^{*3}; \mathrm{gdActionPointOffset}[\mathrm{MT}]\\ &+ \mathrm{ceil}^{*4}(((\mathrm{cChannelIdleDelimiter}[\mathrm{gdBit}] \times \mathrm{gdBitMax}[\mu s] \div 1\mathrm{gdBit}) + \mathrm{gdMaxPropagationDelay}[\mu s])\\ &\div (\mathrm{gdMacrotick}[\mu s] \times (1 - \mathrm{cClockDeviationMax}))) \times 1[\mathrm{MT}]) \quad \cdots\cdots (2)\end{aligned}$$

*1：max(*x*;*y*)は，*x*と*y*の大きいほうを返すことを意味する(例えば，max(1；2)＝2)

*2：gdBGpostEnablePartはバス・ガーディアンを使用しないときは‘0’とすることができる．

*3：マージン

*4：ceil(x)は，xの小数点以下を切り上げることを意味する(例えば，ceil(2.1)＝3)

●静的スロット長

$$\begin{aligned}\mathrm{gdStaticSlot}[\mathrm{MT}] = &\mathrm{gdActionPointOffset}[\mathrm{MT}]\\ &+ \mathrm{ceil}((\mathrm{vStaticFrameLength}[\mathrm{gdBit}] \times \mathrm{gdBitMax}[\mu s] \div 1[\mathrm{gdBit}]) \div (\mathrm{gdMacrotick}[\mu s] \times (1 - \mathrm{cClockDeviationMax}))) \times 1[\mathrm{MT}]\\ &+ \mathrm{vSilencePart}[\mathrm{MT}] \quad \cdots\cdots (3)\end{aligned}$$

図9　静的スロット長を求める計算式

た，二つの連続するフレーム間のチャネル・アイドル境界11gdBit(cChannelIdleDelimiter)もスロット長に必要な項目となります．さらに，バスの長さやスター・カプラによる遅延，クロック精度などを考慮するときには，フレームどうしの衝突を避けるためのスペース(サイレント部分)を定義します．この場合，クラスタ全体の最大伝播遅延(gdMaxPropagationDelay)を考える必要が出てきます．

サイレント部分の長さ(単位はMT)は，**図9**の式(2)で表すことができます．式(1)と式(2)から静的スロット長(gdStaticSlot)を求めることができます(**図9**の式(3))．

では，式(3)から最小の静的スロット長を計算してみましょう．式に入れるパラメータの値はすべて最小値で設定します．すなわち，

- gdActionPointOffset＝1MT
- gdTSSTransmitter＝1gdBit
- gPayloadLengthStatic＝0
- gdBitMax＝0.10015μs(通信速度を10Mbpsとした場合)注1
- gdMaxPropagationDelay ≪ gdMacrotick(クラスタの伝播遅延はマクロティックと比べて十分に小さいので，以下の計算では0とする)
- vStaticFrameLength＝84gdBit

注1：1gdBitはあるネットワークの伝送速度におけるビット(1ビットの伝送時間)を指す．例えば，伝送速度が10Mbpsの場合は1dgBit＝0.1μs(1÷10×10^6)，2.5Mbpsの場合は1gdBit＝0.4μsとなる．ここで，FlexRayの仕様ではクロックの最大誤差は1,500ppm(ppmは100万分の1)と定められている(cClockDeviationMax=±0.0015)．つまり，10Mbpsの場合のgdBitの許容最大値gdBitMaxは0.1μs＋0.1×0.0015μs＝0.10015μsとなる．

注2：細かく言うと，ceil(1.1μs÷(gdMacrotick[μs]×(1－cClockDeviationMax))となるが，ここでは有効数字を小数点以下第2位までとしているので1－cClockDeviationMax＝1として扱っている．

- vSilencePart＝1MT
  ＋ceil(1.1μs÷gdMacrotick[μs])MT 注2

静的スロットの最小時間は公称マクロティック長に依存します．そのため，gdMacrotick＝gdMaxMTNom＝6μsとなります．以上の条件から最小の静的スロット長は，

$$\begin{aligned}\mathrm{gdStaticSlot} &= 2\,\mathrm{MT} + \mathrm{ceil}(8.41\,\mu s \div 6\,\mu s)\,\mathrm{MT}\\ &\quad + \mathrm{ceil}(1.1\,\mu s \div 6\,\mu s)\,\mathrm{MT}\\ &= 2\mathrm{MT} + \mathrm{ceil}(1.41)\,\mathrm{MT} + \mathrm{ceil}(0.18)\,\mathrm{MT}\\ &= 5\mathrm{MT}\end{aligned}$$

となります．

同様の手順で最大の静的スロット長を計算します．式に入れるパラメータの値はすべて最大値で設定します．

- gdActionPointOffset＝31 MT
- gdTSSTransmitter＝13 gdBit
- gPayloadLengthStatic＝127
- gdBitMax＝0.4006μs(通信速度を2.5Mbpsとした場合)
- gdMaxPropagationDelay＝1μs
- vStaticFrameLength＝2636 gdBit
- vSilencePart＝31MT
  ＋ceil(5.41 μs÷gdMacrotick[μs])MT

今度はgdMacrotick＝cdMinMTNom＝1μsとすると，最大の静的スロット長は次のようになります．

$$\begin{aligned}\mathrm{gdStaticSlot} &= 62\,\mathrm{MT} + \mathrm{ceil}(1055.98\,\mu s \div 1\,\mu s)\,\mathrm{MT}\\ &\quad + \mathrm{ceil}(5.41\,\mu s \div 1\,\mu s)\,\mathrm{MT}\\ &= 62\,\mathrm{MT} + \mathrm{ceil}(1055.98)\,\mathrm{MT}\end{aligned}$$

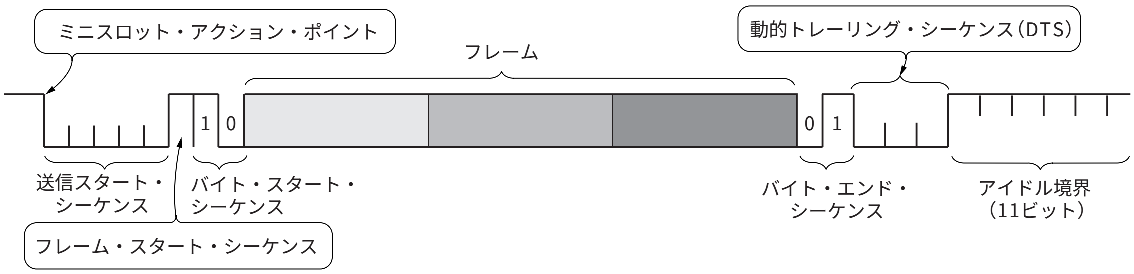


図10 動的スロットのフレーム長

$$+\mathrm{ceil}(5.41)\,\mathrm{MT} = 1124\ \mathrm{MT}$$

以上のことから，静的スロット長は仕様で規定されている範囲5～2,047MT(gdStaticSlot=5-2047)に入っていることがわかります．

前回は静的スロット幅を75MTと定義しました．この静的スロットにおいて，ペイロード長をどの程度設定できるか計算してみましょう．ここで，各パラメータの値は上記の最小の静的スロット長を算出した際と同じものとします．ペイロード長をXとおいて，式(1)に当てはめます．

$$\mathrm{vStaticFrameLength} = 1+1+80+X\times 20+2 = (84+20X)\,\mathrm{gdBit}$$

次に，gdMacrotick=1μsと設定して，式(2)からサイレント部分を求めます．

$$\mathrm{vSilencePart} = 1\mathrm{MT}+\mathrm{ceil}(1.1\,\mu\mathrm{s}\div 1\,\mu\mathrm{s})\,\mathrm{MT} = 3\mathrm{MT}$$

これらの値と静的スロット幅75MTを式(3)に当てはめると，ペイロード長が求まります．

$$75\mathrm{MT} = 4\mathrm{MT}+\mathrm{ceil}((84+20X)\times 0.1\times 1\mathrm{MT})$$
$$71\mathrm{MT}=8.4\mathrm{MT}+2.0X\mathrm{MT}$$
$$X = 32\mathrm{MT}$$

前回，1MT＝1μsと設定しました．そのため，ペイロード長は32μsとなります．通信速度が10Mbpsでは1gdBit＝0.1μsなのでペイロード長は32バイトになります．

● 送信開始と終了でアクション・ポイントを使用

動的セグメントのミニスロットの構成は，静的セグメントの場合と多少異なります．何が違うかというと，**図10**に示すようにフレーム・エンド・シーケンスの後に動的トレーリング・シーケンス(DTS：Dynamic Trailing Sequence)が追加されていることです．

ミニスロットでフレームを送信するタイミングは，静的スロットと同じようにアクション・ポイントですが，静的スロットのアクション・ポイントと区別するためにここではミニスロット・アクション・ポイントと呼びます．ミニスロットの場合，送信の終了もミニスロット・アクション・ポイントで終了します．つまり，1ミニスロットの中に送信の開始のミニスロット・アクション・ポイントと送信の終わりのミニスロット・アクション・ポイントがあるということです．

送信の終わりのミニスロット・アクション・ポイントのタイミングがこの動的トレーリング・シーケンスで実行されます．また，早すぎるチャネル・アイドルの検出を防止します．動的トレーリング・シーケンスの範囲は2～gdMinislot＋2(gdBit)と定められています．

＊　＊　＊

今回は，1静的スロットにどの程度データを送信できるのか(ペイロード長)を求めてみました．その結果，正味のデータの占める割合は1静的スロット当たりだいたい43％程度であることがわかりました．大量にデータを扱う場合は，ペイロード長をとくに考慮に入れる必要があります．

次回は，受信側から見たときのフレームについてお話しします．

さとう・みちお
フリースケール セミコンダクター ジャパン(株)

<筆者プロフィール>
佐藤道夫．1982年にモトローラに入社し，半導体事業部に所属．品質保証や営業を経験し，1992年からコントロール&ネットワークのアプリケーション・エンジニアとして，ロンワークスの普及に従事する．現在は，フリースケール セミコンダクタージャパンの半導体セクタートランスポーテーション&スタンダードプロダクツグループにおいて，おもに車載向けのネットワークの技術担当マネジャーをしている．